

BAB III

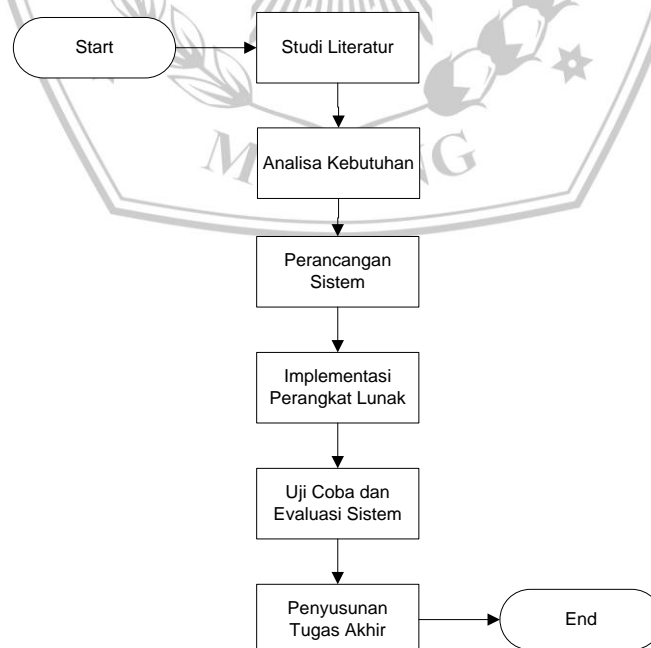
METODOLOGI PENELITIAN

3.1. Deskripsi Umum

Pada penelitian ini akan dilakukan analisa terhadap kinerja *routing Content Centric Network (CCN)* yang diimplementasikan dalam jaringan multimedia *streaming* atau *video streaming*. Parameter yang akan diuji adalah parameter *QoS (Quality of Service)* seperti, *delay/latency*, *jitter*, *throughput*, dan *packet loss*. Selain itu pada penelitian ini juga akan dilakukan analisa untuk mengetahui kemampuan *Content Centric Network (CCN)* saat proses pengiriman data *video streaming* pada sistem operasi yang berbeda-beda, yaitu sistem operasi *linux* sebagai penyedia layanan *streaming*, dan sistem operasi *linux*, *windows*, dan *android* sebagai sisi *client*, serta membandingkan kinerja sistem yang telah diimplementasikan *Content Centric Network (CCN)* dengan sistem tanpa *Content Centric Network (CCN)*.

3.2. Alur Penelitian

Didalam penelitian ini terdapat alur penelitian yang bertujuan untuk memudahkan penulis untuk memberikan hasil analisa yang terstruktur dan jelas agar dapat memberikan informasi hasil analisa dengan baik.



Gambar 3.1. Flowchart Alur Penelitian

Pada gambar 3.1 menjelaskan tentang alur rancangan penelitian yang akan dilakukan peneliti. Di mulai dari alur penelitian studi literatur yang digunakan adalah seperti beberapa jurnal, *e-book*, artikel ilmiah, serta beberapa sumber dari internet untuk menambah wawasan tentang semua yang berhubungan dengan penelitian, selanjutnya analisa kebutuhan. Untuk analisa kebutuhan terdiri pengumpulan data dan identifikasi masalah, Alur penelitian yang selanjutnya tentang perancangan sistem, yang menjelaskan tentang gambaran umum sistem dan arsitektur sistem. Kemudian implementasi perangkat lunak menjelaskan tentang realisasi sistem yang telah dirancang dan meimplementasikan perangkat lunak yang digunakan. Pada tahap uji coba dan evaluasi menjelaskan tentang skenario-skenario yang akan di ujikan pada sistem. Setelah itu langkah terakhir adalah penyusunan laporan tugas akhir oleh penulis.

3.3. Analisis Kebutuhan Sistem

Analisa terhadap kebutuhan sistem dilakukan pada tahap ini. Untuk analisa kebutuhan terdiri pengumpulan data dan identifikasi masalah. Sistem ini terbagi menjadi 3 komponen utama yaitu penyedia layanan (*streaming server*), jaringan (*network*), serta sisi *client*. Sistem ini menggunakan jaringan lokal untuk mengetahui kehandalan dari proses *streaming* yang akan berlangsung nantinya.

3.3.1. Analisis Kebutuhan Hardware dan Software

Analisa kebutuhan *hardware* dan *software* menggambarkan kebutuhan perangkat yang akan menjadi wadah sistem dalam mengelola data aplikasi. Kebutuhan *hardware* dan *software* dalam membangun sistem ini adalah :

a. Kebutuhan Perangkat Lunak

Terdapat beberapa perangkat lunak yang digunakan dalam membangun sistem ini adalah sebagai berikut :

Tabel 3.1. *Software* server layanan *streaming*

<i>Software</i> / Perangkat Lunak
VLC Player
CCNx
Ubuntu 16.04.6 LTE

b. Kebutuhan Perangkat Keras

Perangkat keras yang digunakan selama membangun sistem ini adalah dengan spesifikasi sebagai berikut :

Tabel 3.2. Spesifikasi *hardware* server layanan *streaming*

<i>Processor</i>	Intel Core i5-4570 CPU @3.20GHz
<i>Memory (RAM)</i>	8GB
<i>Operating System</i>	Ubuntu 16.04.6 LTS
<i>Harddisk</i>	500GB

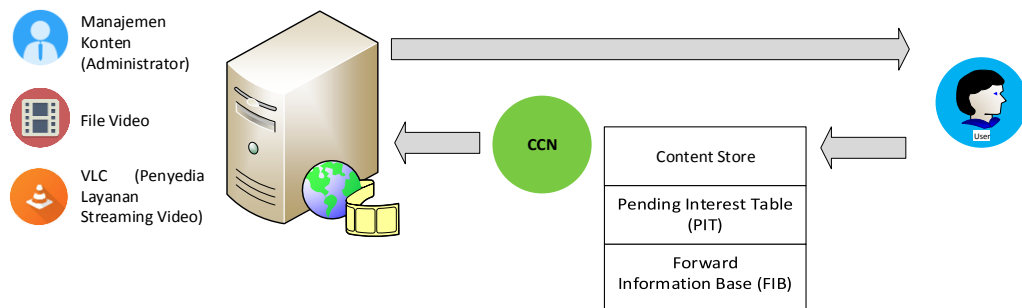
3.3.2. Analisis Sistem Sisi Streaming server

Sistem pada sisi *streaming* server ini dibangun pada *operating system* berbasis *Linux Ubuntu* yang merupakan salah satu distribusi sistem operasi *linux* yang paling sering di gunakan, Sistem operasi *linux ubuntu* yang digunakan memiliki versi *Linux Ubuntu 16.04.6 LTS*. Penggunaan distro *linux* ini dikarenakan kebutuhan sistem untuk menjalankan program yang lebih kecil dengan kata lain lebih sedikit dalam konsumsi perangkat lunak/*software* sehingga dimungkinkan dalam proses *streaming* nantinya dapat bekerja dengan maksimal.

Proses yang terjadi pada sistem ini, dimulai dengan penyediaan *content* multimedia oleh *streaming server*. Pada proses pembuatan *streaming server* ini dibutuhkan komponen pendukungnya yaitu *Video Lan Client (VLC)* digunakan untuk media *streaming server*. Dari *streaming server* ini kemudian akan didistribusikan melalui sebuah jaringan lokal yang telah diimplementasikan sistem *routing Content Centric Network (CCN)* dan dipakai untuk proses transfer data.

3.3.3. Analisis Sistem Sisi Jaringan

Sistem pada sisi jaringan ini adalah sebuah sistem dimana terjadinya proses transfer data multimedia dari *streaming server* menuju ke *client* sehingga layanan yang telah disediakan oleh *streaming server* dapat diakses oleh sisi *client*. Pada tahap ini akan kita ketahui kehandalan dari layanan *routing Content Centric Network (CCN)* ini sampai pada *client* apakah ada penurunan atau tidaknya dari fitur *streaming* multimedia yang diakses.

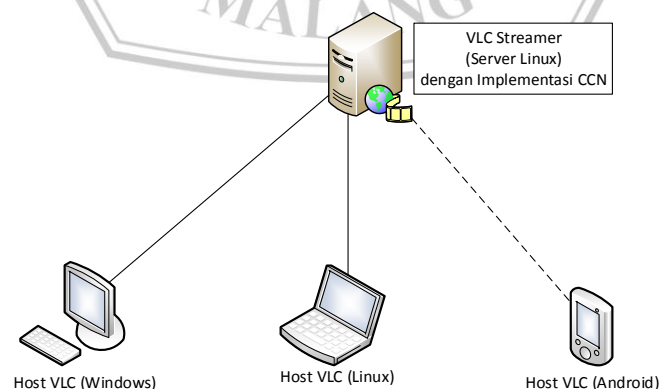


Gambar 3.2. Sistem Sisi Jaringan

Pada Gambar 3.2 *Client* akan melakukan *request video streaming* dengan sebuah *name file* tertentu yang telah disediakan oleh *streaming server*. Kemudian, perangkat akan mengirimkan paket *interest* ke dalam jaringan CCN. Pada jaringan CCN akan mengecek paket *interest* yang diterima. Jika pada *content store* tersedia maka secara langsung konten akan dikirimkan menuju asal dari paket *interest*. Jika tidak ada, maka pada *node server* tersebut akan mengecek *Pending Interest Table (PIT)* untuk mengecek jika ada *request* untuk konten yang sama. *PIT* akan mengumpulkan semua *request* yang sama dan melanjutkan proses *request* ke *node server streaming* yang lain (jika tersedia).

3.4. Arsitektur Sistem

Pada bagian ini merupakan gambaran tentang arsitektur sistem yang akan dibangun. Gambar 3.3 akan menunjukkan gambaran arsitektur sistem :



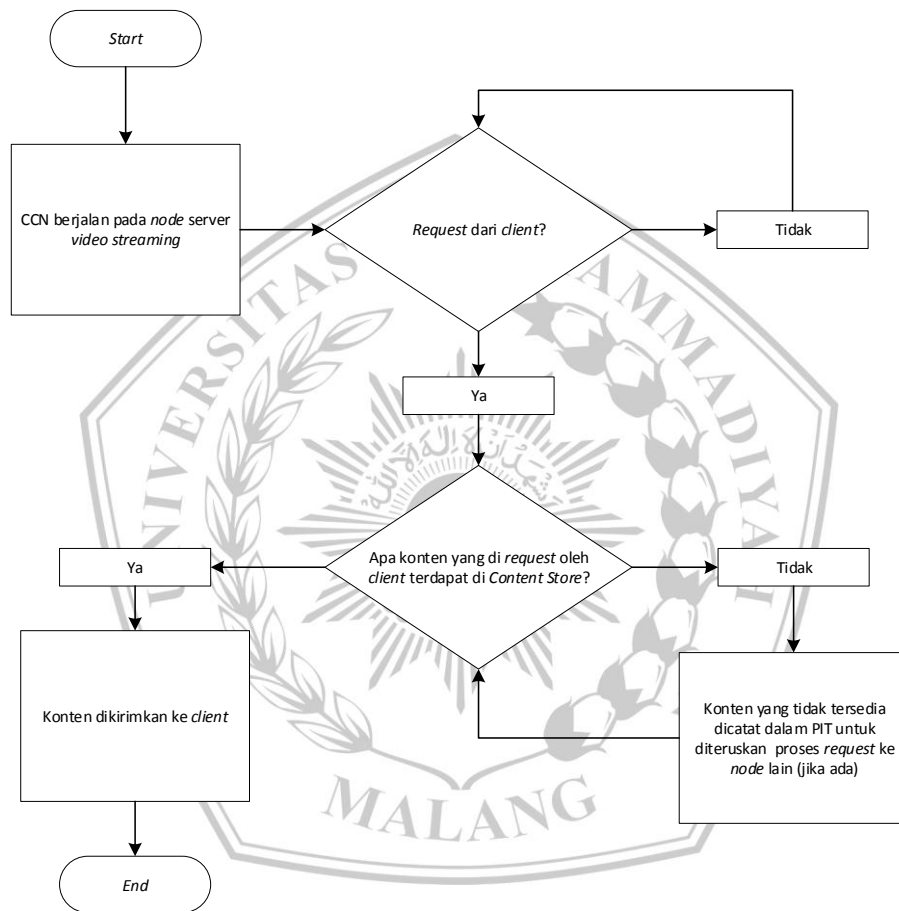
Gambar 3.3. Arsitektur Sistem

Pada gambar 3.3 di atas menjelaskan arsitektur sistem. Pada *server* diimplementasikan sistem *routing CCN* sebagai sistem komunikasi pada video

streaming. Server bertindak sebagai *content store* yang menyediakan file video dan telah dikonfigurasi oleh *administrator* penyedia layanan *video streaming*. Aplikasi penyedia *video streaming* menggunakan *VLC Player*.

3.5. Diagram Alur Sistem

Pada bagian ini merupakan gambaran tentang alur kerja / *flowchart* sistem yang akan dibangun. Gambar 3.4 akan menunjukkan gambaran alur kerja sistem :



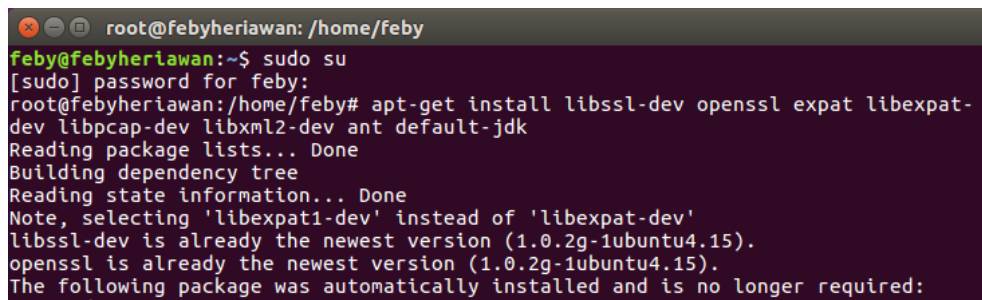
Gambar 3.4. Diagram alur sistem CCN

Gambar 3.4 menunjukkan diagram alir dari proses kerja sistem CCNx. Pertama-tama CCN daemon pada node server berjalan. Jika salah satu client pada jaringan CCN melakukan request, maka request tersebut akan dilanjutkan menuju server video streaming. Kemudian dilakukan pengecekan apakah pada content store tersebut terdapat data yang diinginkan oleh client. Jika ada, maka data akan dikirimkan kembali kepada client yang mengirimkan request. Sedangkan jika data yang diinginkan

tidak ada pada *content store*, maka *request* akan di catat dalam bentuk *cache* pada *PIT* yang kemudian diteruskan ke *node server* lainnya (jika tersedia *node server video streaming* lain).

3.6. Instalasi Tools dan Aplikasi Sistem

Pada tahap ini dilakukan instalasi *CCNx*, untuk menunjang implementasi *CCNx* berjalan dengan baik pada server juga di install beberapa dependensi.



```

root@febyheriawan: /home/feby
feby@febyheriawan:~$ sudo su
[sudo] password for feby:
root@febyheriawan:/home/feby# apt-get install libssl-dev openssl expat libexpat-
dev libpcap-dev libxml2-dev ant default-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'libexpat1-dev' instead of 'libexpat-dev'
libssl-dev is already the newest version (1.0.2g-1ubuntu4.15).
openssl is already the newest version (1.0.2g-1ubuntu4.15).
The following package was automatically installed and is no longer required:

```

Gambar 3.5. Printah instalasi dependensi tambahan

Kemudian install aplikasi *VLC Player* yang akan dikonfigurasi dengan implementasi *CCNx* agar dapat saling terhubung. Terdapat *plugin* pada *CCNx* yang harus disalin ke *library plugin VLC Player* agar video dapat pada repositori *CCNx* dapat diputar oleh *VLC Player*. Sebagai software analisa jaringan dilakukan instalasi *Wireshark*.

3.7. Pengujian Skenario dan Pengambilan Hasil

Pada tahap ini akan dilakukan pengujian sistem yang telah dibuat. Pengujian terdiri dari beberapa skenario yang dijalankan untuk mengetahui kinerja *CCNx* dengan melakukan pembebanan terhadap jaringan yang telah dibuat, dan untuk mengetahui waktu *delay*, *throughput*, *jitter*, dan *packet loss streaming* video dengan implementasi *CCN*. Dalam pengujian ini akan dilakukan pada 3 sistem operasi yaitu *Windows*, *Linux*, dan *Android*.

Tabel 3.3. Hasil pengujian implementasi *CCN* pada *Client OS Linux*

<i>Linux</i>			
<i>Throughput</i>	<i>Jitter</i>	<i>Delay</i>	<i>Packet Loss</i>

Tabel 3.4. Hasil pengujian implementasi *CCN* pada *Client OS Windows*

<i>Windows</i>			
<i>Througput</i>	<i>Jitter</i>	<i>Delay</i>	<i>Packet Loss</i>

Tabel 3.5. Hasil pengujian implementasi *CCN* pada *Client OS Android*

<i>Android</i>			
<i>Througput</i>	<i>Jitter</i>	<i>Delay</i>	<i>Packet Loss</i>

Kemudian akan dibandingkan sistem yang telah diimplementasikan *CCN* dengan sistem yang tidak diimplementasikan dengan *CCN*. Berikut merupakan tabel pengujian perbandingan implementasi sistem *CCN* dan tanpa *CCN* di beberapa sistem operasi yang berbeda :

Tabel 3.6. Hasil perbandingan pengujian implementasi *CCN* dan tanpa *CCN* pada *Client OS Linux*

<i>Linux</i>							
<i>CCN</i>				<i>Tanpa CCN</i>			
<i>Througput</i>	<i>Jitter</i>	<i>Delay</i>	<i>Packet Loss</i>	<i>Througput</i>	<i>Jitter</i>	<i>Delay</i>	<i>Packet Loss</i>

Tabel 3.7. Hasil perbandingan pengujian implementasi *CCN* dan tanpa *CCN* pada *Client OS Windows*

<i>Windows</i>							
<i>CCN</i>				<i>Tanpa CCN</i>			
<i>Througput</i>	<i>Jitter</i>	<i>Delay</i>	<i>Packet Loss</i>	<i>Througput</i>	<i>Jitter</i>	<i>Delay</i>	<i>Packet Loss</i>

Tabel 3.8. Hasil perbandingan pengujian implementasi *CCN* dan tanpa *CCN* pada *Client OS Android*

<i>Android</i>							
CCN				Tanpa CCN			
<i>Througput</i>	<i>Jitter</i>	<i>Delay</i>	<i>Packet Loss</i>	<i>Througput</i>	<i>Jitter</i>	<i>Delay</i>	<i>Packet Loss</i>

